Leading Change from the Test Team

John Ruberto – Concur, Inc. – Director of Mobile QE

References

bit.ly/JRChange

First, a story

Tell me about your title

Test Manager

- Focus is testing
- Organizing work around testing

Quality Leader

- Focus is whole life-cycle & process that leads to software delivery
- Influencing outcomes

Most Changes Efforts Fail

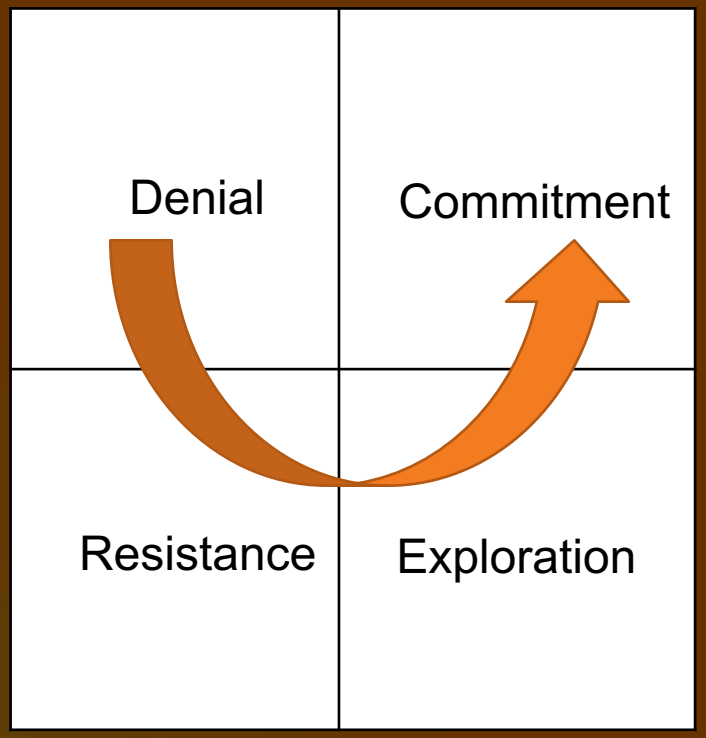
Why changes fail

Kotter: Leading Change – Why Transformation Efforts Fail, Harvard Business Review

1. Not establishing a great enough sense of urgency
2. Not creating a powerful enough guiding coalition
3. Lacking a vision
4. Under communicating the vision by a factor of 10
5. Not removing obstacles to the new vision
6. Not systematically planning, creating, short-term wins
7. Declaring victory too soon
8. Not anchoring changes into corporate cultures

Change is Hard

The Change Curve



Kübler-Ross model

1. Denial
2. Anger
3. Bargaining
4. Depression
5. Acceptance

Ref: MindTools, DREC, etc.

Tale of 2 changes

- Process / SDLC Change
  - Moving from 6-weeks to monthly release cycles

- Technology Change:
  - Institute static analysis in our development methodology

Roles in Change Leadership

- Change Leader
  - Leads in generating the vision
  - Drives the overall process for change

- Change Agent
  - Active Participant
  - “First Follower”
  - Video

Change leadership in 4 steps

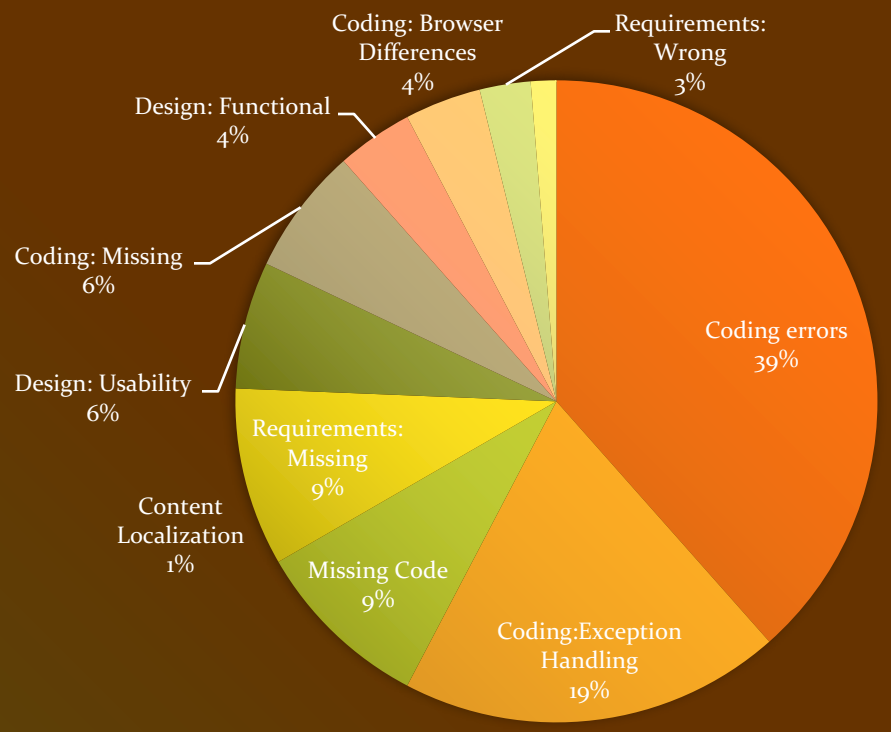
1. Build the case for change
2. Plan the change
3. Test the change
4. Rollout and make adjustments

Build the case for change

- Identify the need
  - Retrospectives
  - Benchmark other organizations
  - Defect escapes / escalations
  - Industry trends / best practices
  - Ideas from team, innovations, experiments
  - Root cause analyses
  - Risk analysis

Build the case for change

Example: Defects found in system test



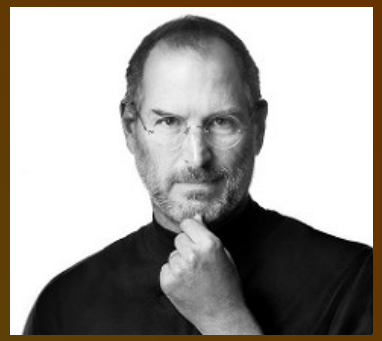
- Root cause analysis
  - 70% of errors found in system test were caused by coding errors
  - How can we improve? (brainstorm)
  - Idea: start doing static analysis to find errors earlier

Build the case for change

- Think about your audience
  - Developers
  - Testers
  - Product Managers / Product Owners
  - Project Managers / Scrum Masters
  - Leaders

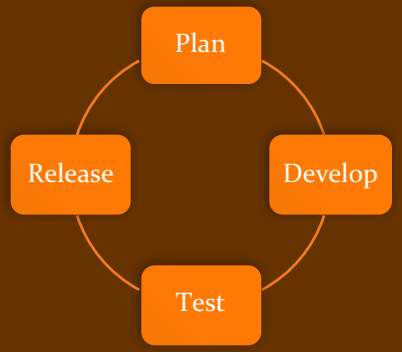
Communicate in the language of your audience

- Ideas -> Vision , Features -> Benefits



MP3 player
With small HDD

1000 songs
In my pocket



Faster release cycles

Quicker feedback
From customers

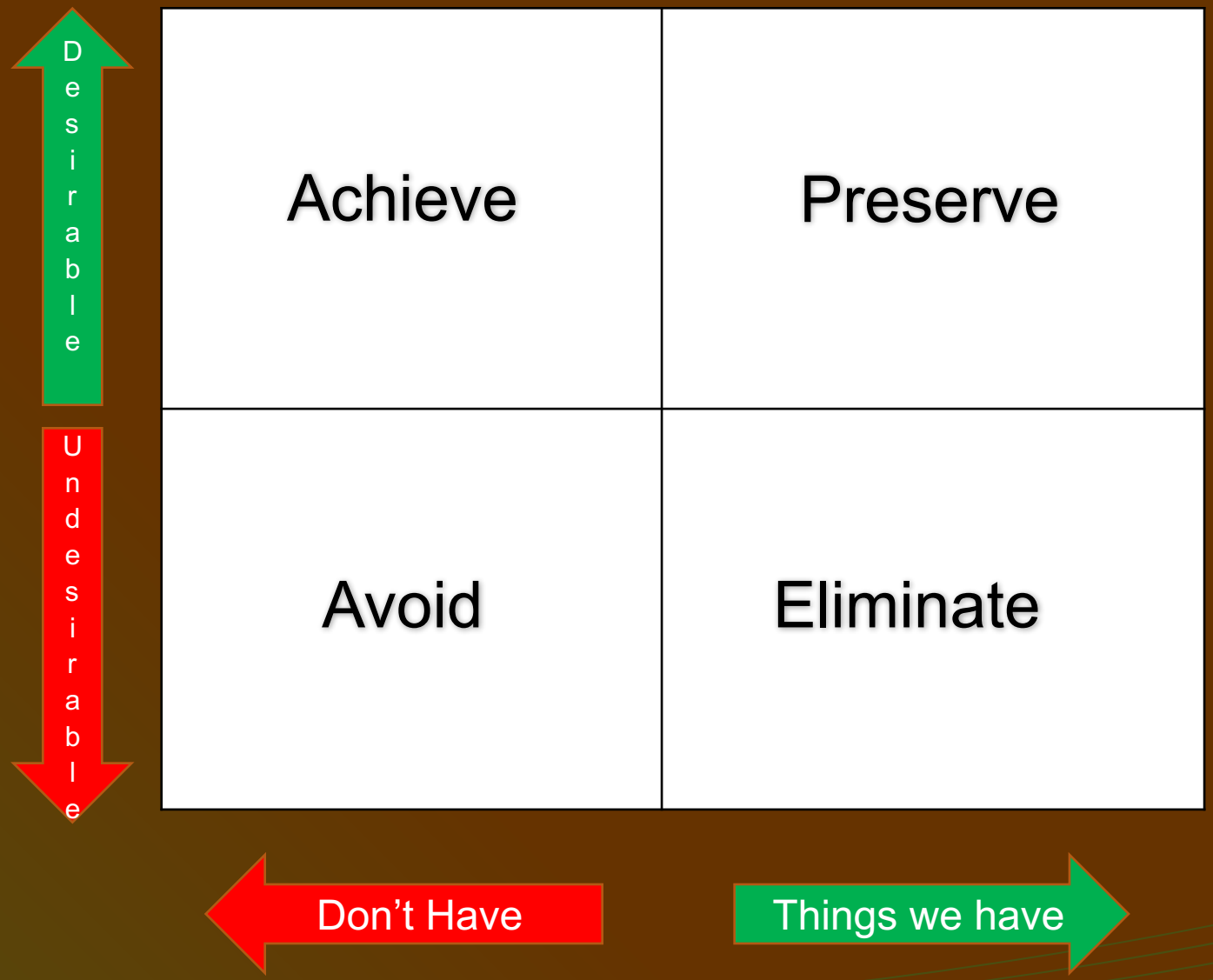
Static analysis checks
As part of the build

Automated code-
Review buddy

Plan the change

- Communicate, communicate, communicate
  - Organization’s goals
  - 1:1 sessions , Team meetings, all-hands
  - Identify people who are:
    - Impacted by the change
    - Change agents
    - Passionate defenders of the status quo
  - Ask:
    - What needs to change to make the vision reality?
    - What is working that we need to preserve?
    - What worries you?

The Goals Grid



The Goals Grid

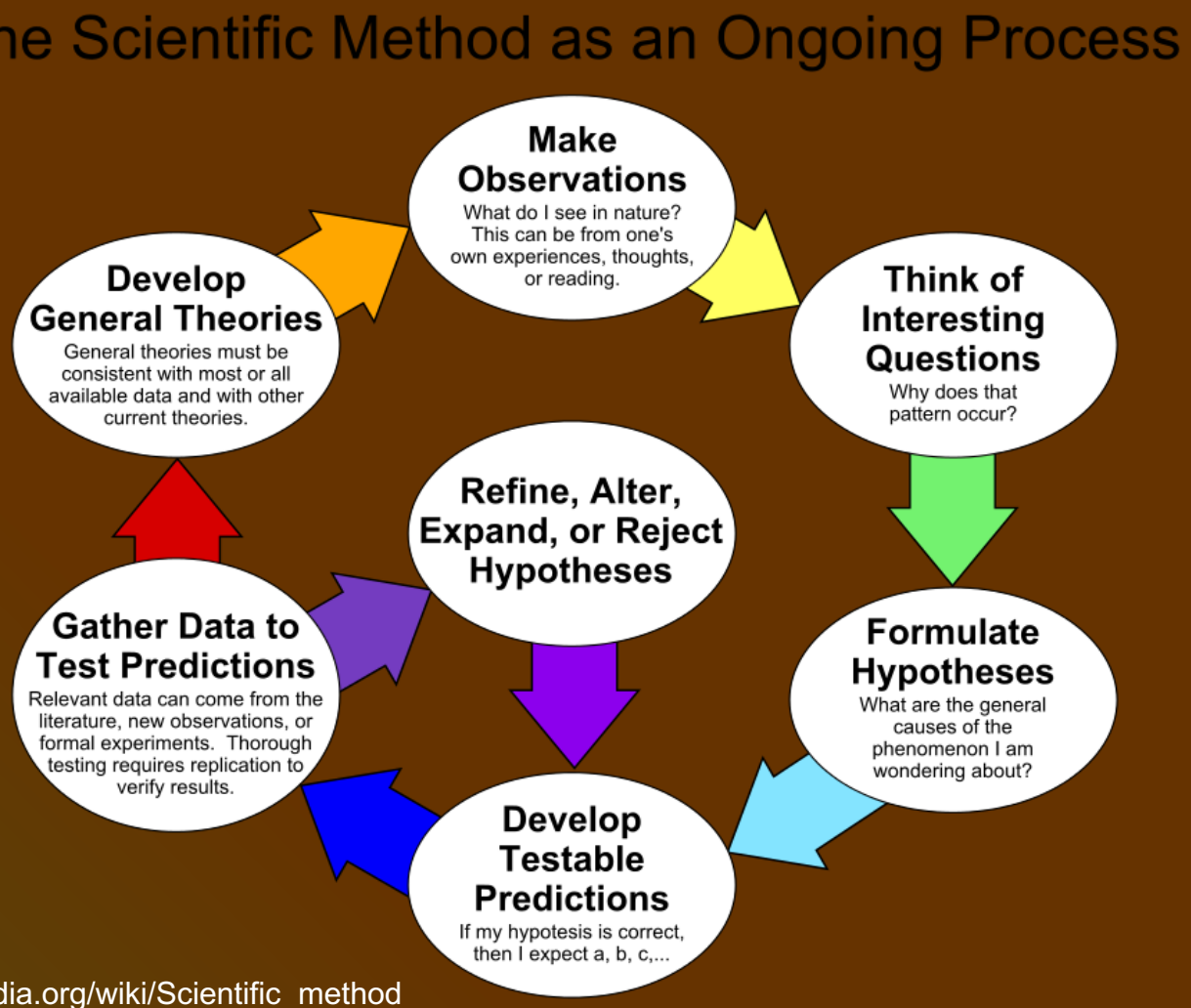
Desirable	**Achieve** Firm completion milestone Regression in only 1 week Automated deployment Definition of Done = "Customer ready" Better content communication across functions (ops, support, etc.)	**Preserve** Demos with support team before release Staged deployment with subset of customers Feature configuration switches Level of documentation (requirements, design, etc.)
Undesirable	**Avoid** Keeping the same practices, only compressing time and making people work OT Embarrassing bugs Double the manual deployments across data centers Using "looming deadlines" to encourage urgency	**Eliminate** "This feature must release on this date" mindset Late scrambles for design activities. Pre-prod test environments instability

Don't Have

Things we have

Communicate, Communicate, Communicate

Break out the scientific method

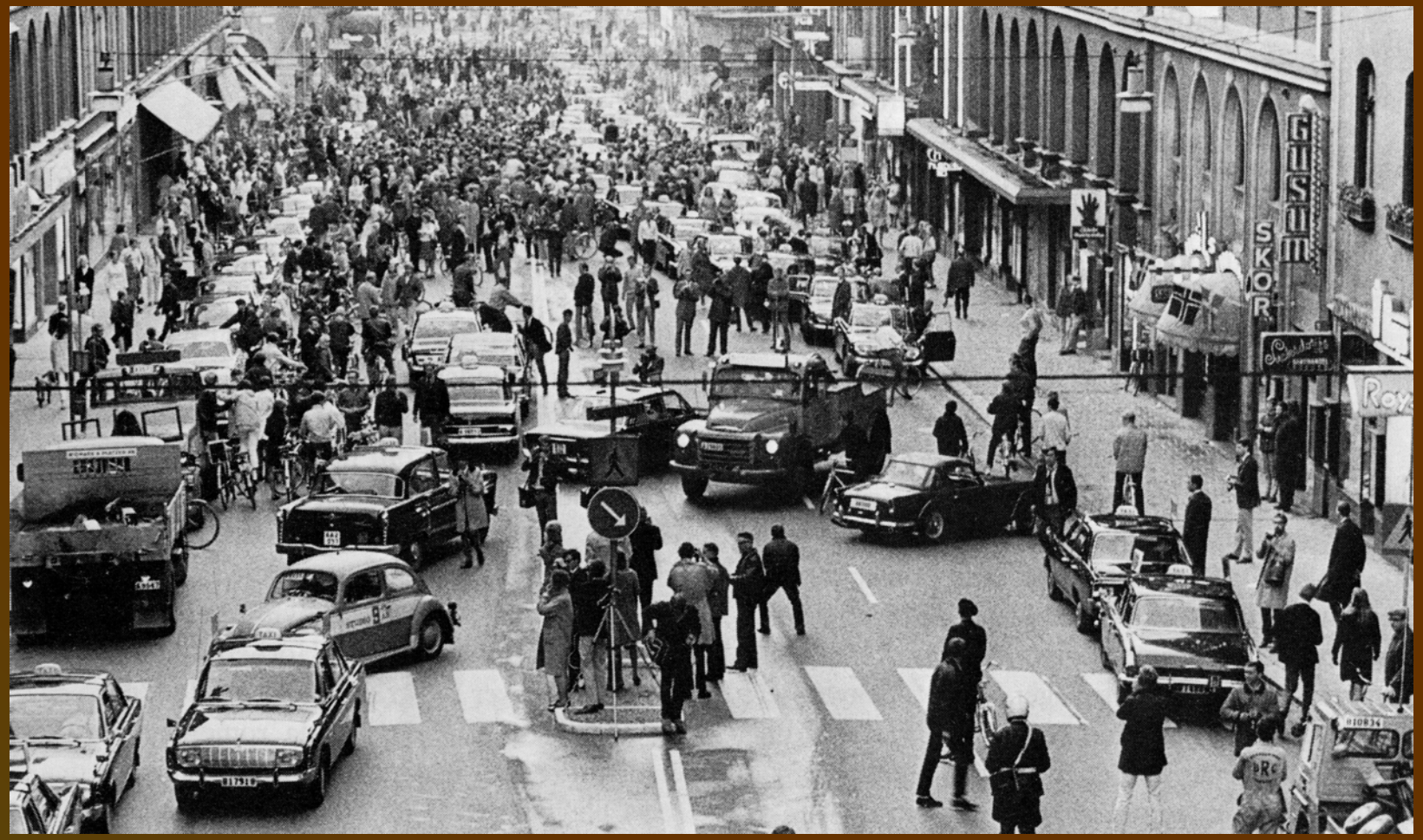


https://en.wikipedia.org/wiki/Scientific_method

Test the change

	Faster releases	Static Analysis
Testable prediction	We can complete system test in 1 week if we use the new milestone definition	A tool exists that will find valid/meaningful coding errors with minimal false positives
Experiment	Use new milestone, keep 2 weeks in schedule for system test, but try to complete in 1	Identify a list of tools, and try out the most promising. Compare results
Refine	Locks on change control	Found 2 tools that worked well together (complementary results)

Rollout and Make Adjustments

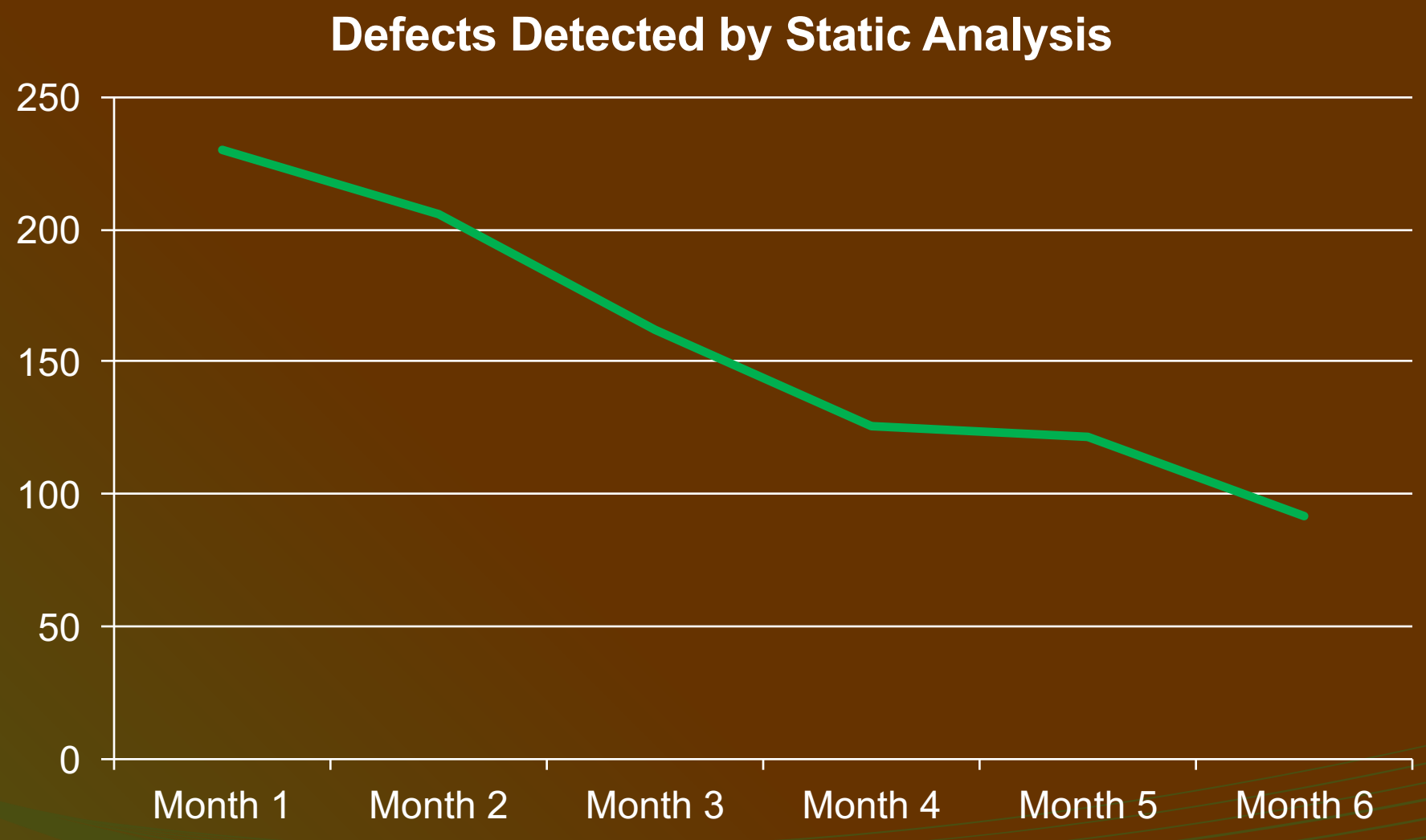


Högertrafikomläggningen - Dagen H (Sept 3rd, 1967)

Be Persistent

- Actively seek out risks and issues
- Communicate openly about risks & progress (often)
- Ask for feedback: How can we improve this plan?

Show progress



Summary

- Building the burning platform is most important
  - Speak the language of your stakeholders
  - Articulate your vision
  - “be in love with the problem, not your solution”
- Communicate, Communicate, Communicate
  - Before, During, and After
- Persist when it gets difficult
  - Are you still solving the original problem?
    - Otherwise, pivot to a different solution

Thank You!

Please leave a feedback card

Links to materials referenced are here: bit.ly/JRChange

About Me

@JohnRuberto

blog.ruberto.com

linkedin.com/in/ruberto

About Concur

concur.com/careers

Travel & Expense Management w/ 35m active users

In top 50 best places to work (glassdoor.com)

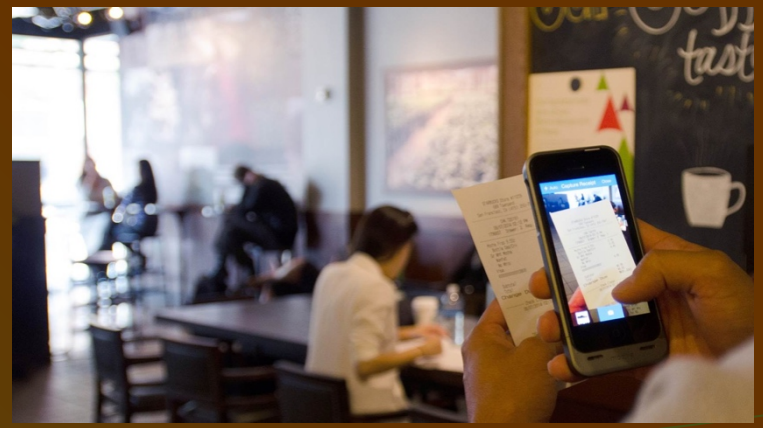


Image Credits

- Slide 2: *yonamaro from wikimedia*
- *Slide 9: US Coast Guard (public domain)*
- Slide 13: (jobs) segagman from flicker creative commons
- Slide 13: (code) Pixabay (public domain)
- Slide 17: McDonalds Inc. (public domain)
- Slide 20: Jan Collsioo (public domain)
- Slide 24: Concur, Inc. (with permission)